UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/710,218	11/10/2000	Timothy L. Harris	1004-4896	4731

42714 7590 01/26/2006

ZAGORIN O'BRIEN GRAHAM LLP (004)
7600B NORTH CAPITAL OF TEXAS HIGHWAY
SUITE 350
AUSTIN, TX 78731-1191

EXAMINER	
ZHEN, LI B	

ART UNIT	PAPER NUMBER
2194	

DATE MAILED: 01/26/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/710,218	HARRIS, TIMOTHY L.	
	Examiner	Art Unit	
	Li B. Zhen	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 October 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 – 30 are pending in the application.

2. In view of the Appeal Brief filed on 10/31/2005, PROSECUTION IS HEREBY REOPENED. New grounds of rejection are set forth below.

To avoid abandonment of the application, appellant must exercise one of the following two options:

(1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37 CFR 1.113 (if this Office action is final); or,

(2) initiate a new appeal by filing a notice of appeal under 37 CFR 41.31 followed by an appeal brief under 37 CFR 41.37. The previously paid notice of appeal fee and appeal brief fee can be applied to the new appeal. If, however, the appeal fees set forth in 37 CFR 41.20 have been increased since they were previously paid, then appellant must pay the difference between the increased fees and the amount previously paid.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. Claims 1 – 30 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. The independent claims recite: "wherein the first synchronization primitive atomically examines and updates a single target, the updating being conditional on the examination" [claim 1, lines 8-9; claim 16, lines 7-8; claim 25, lines 14-16; claim 29, lines 9-11]. According to the applicant's arguments [appeal brief dated 10/31/2005, p. 2, lines 13 – 15] and the claims [i.e. claims 3, 21 and 30], the first synchronization primitive is a compare-and-swap (CAS) or a locked/store-

and compare (LL/SC) instruction pair. However, the specification also discloses that `remove` operation with CAS operation C3 [Fig. 5] fails to remove any node with a key greater than or equal to a value *x* [*x=10* in the example on p. 31, lines 5 – p. 32, line 4 of the specification]. In order to address this problem, the specification discloses replacing the CAS operation (C3) of Fig. 5 with an atomic double-compare-and-swap (DCAS) operation [p. 32, lines 5 – 13, emphasis added] in order to remove any node with a key greater than or equal to a value *x* [p. 31, lines 5 – p. 32, line 4 of the specification]. The DCAS operation compares two targets [p. 32, lines 10 – 13 of the specification]. The independent claims recite "wherein the first synchronization primitive atomically examines and updates a single target, the updating being conditional on the examination" while, the specification requires the first synchronization primitive to compare two targets [p. 32, lines 10 – 13 of the specification]. Therefore, the applicant fails to disclose the all the recited limitations in the specification as filed.

5. Claims 1 – 30 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Specifically, the specification does not disclose how to replace a compare-and-swap (CAS) operation with double-compare-and-swap operation (DCAS) and how to replace a compare-and-swap operation with a locked load/store-and-compare (LL/SC) instruction pair. Applicant's specification discloses replacing a CAS operation with a DCAS operation [p. 32, lines 5 – 13]. The specification simply discloses replacing a CAS operation with a DCAS operation and does not provide details on how to convert a CAS-based implementation to a DCAS-based implementation. In addition, applicant submitted that the idea that a DCAS operation may simply be replaced with a CAS flies in the face of nearly 20 years of research in the art [p. 6, lines 22 – 24 of the appeal brief submitted 10/31/2005] and one cannot merely replace a DCAS with a CAS operation while maintaining any semblance of a pre-existing algorithmic approach to managing concurrency because CAS-based algorithm typically requires a different

concurrency management techniques [p. 6, line 29 – p. 7, line 3 of the appeal brief submitted 10/31/2005]. The specification does not disclose how to change the concurrency management technique to support a DCAS-based implementation or a hybrid implementation [supporting both CAS and DCAS operations]. Furthermore, the specification does not provide a detailed process on how to replace a compare-and-swap operation with a locked load/store-and-compare (LL/SC) instruction pair. The specification simply notes that other primitives such as a locked load/store-and-compare (LL/SC) instruction pair may be employed in some variations [p. 6, lines 25 – 30]. The arguments for replacing a compare-and-swap (CAS) operation with double-compare-and-swap operation (DCAS) also apply to replacing a compare-and-swap operation with a locked load/store-and-compare (LL/SC) instruction pair. Moreover, Jayanti submits that no existing multiprocessor system supports the LL/SC instructions in hardware ["Efficient and Practical Constructions of LL/SC Variables," Jayanti et al., p. 285, Abstract]. Indeed, applicant's specification describes hardware support for "test-and-set", "fetch-and-add" and "compare-and-swap" primitives [p. 8, lines 25 – 31] but did not provide hardware support for LL/SC synchronization instructions. The specification does not implicitly or inherently disclose how to replace a compare-and-swap (CAS) operation with double-compare-and-swap operation (DCAS) and how to replace a compare-and-swap operation with a locked load/store-and-compare (LL/SC) instruction pair.

Claim Rejections - 35 USC § 101

6. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

7. Claim 28 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

8. Claim 28 is not limited to tangible embodiments. In view of Applicant's disclosure, the medium is not limited to tangible embodiments, instead being defined as

including both tangible embodiments (e.g., disk, tape or other magnetic storage medium; claim 28, lines 3-4) and intangible embodiments (e.g., wireless communications medium; claim 28, line 4). As such, the claim is not limited to statutory subject matter and is therefore non-statutory. To overcome this type of 101 rejection the claims need to be amended to include only the physical computer media and not a transmission media or other intangible or non-functional media.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. **Claims 1 – 8, 10 – 22 and 25 – 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over "A Lock-Free Multiprocessor OS Kernel" [hereinafter Massalin] in view of U.S. Patent No. 6,651,146 to Srinivas et al. [hereinafter Srinivas], both references cited in the previous office action.**

11. As to claim 16, Massalin teaches the invention substantially as claimed including managing access to a linked-list of nodes susceptible to concurrent operations on a group [Compare-and-Swap to perform linked-list insert and delete from the head of the list, Section 3.3, p. 5, second paragraph; in the middle of the list, we can achieve the same effect by deleting a node only when the permanence of the previous node is guaranteed; Section 3.3, p. 6], comprising:

separating deletion of a value from the group into at least two functional sequences [do this in two steps: (1) mark the nodes to be deleted and leave them in the list (2) if the previous node is not marked for deletion, sit on it and delete the original node marked for deletion; p. 6, Section 3.3];

the first function sequence performing a logical deletion of the value using a synchronization primitive [set the mark at the mark at the same time we enter the node using a two-word Compare-and-Swap; Section 3.3, p. 6] to mark a corresponding one of the nodes [(1) mark the nodes to be deleted and leave them in the list; p. 6, Section 3.3]; and

the second functional sequence excising the marked node from the linked-list [(2) if the previous node is not marked for deletion, sit on it and delete the original node marked for deletion; p. 6, Section 3.3].

12. Although Massalin teaches the invention substantially, Massalin does not specifically teach the synchronization primitive atomically examines and updates a single target, the updating being conditional on the examination.

However, Srinivas teaches managing access to a linked-list of nodes susceptible to concurrent operations on a group [col. 2, line 65 – col. 3, line 14], deleting a value from the group [remove elements from a list; col. 6, lines 1 – 18], the first function sequence performing a logical deletion of the value using a synchronization primitive [removal or data elements from the front of the list using an atomic operation of CAS in place of locks; col. 4, lines 13 – 28] to mark a corresponding one of the nodes [CAS (first_ptr, cur_first, new) of step 407, the first_ptr and cur_first compare and the first _ptr is set equal to the variable "new" the next pointer of the current first data element....As a result of step 407, the _current first element is no longer pointed to and thus is removed from the list; col. 12, lines 28 – 67] and the synchronization primitive atomically examines [CAS, as an atomic operation; col. 8, lines 3 – 13] and updates a single target, the updating being conditional on the examination [col. 7, line 60 – col. 8, line 3].

13. It would have been obvious to a person of ordinary skill in the art at the time of the invention to apply the teaching of a synchronization primitive that atomically examines and updates a single target, the updating being conditional on the examination as taught by Srinivas to the invention of Massalin because primitive operations including an atomic operation of "compare and swap" (CAS) allow list management without the use of locks [col. 5, lines 57 – 67 of Srinivas].

14. As to claim 1, Massalin as modified teaches non-blocking [Compare-and-swap is the foundation of lock-free synchronization; Section 2.3, p. 2 of Massalin] concurrent shared object representation [Two-word Compare-and-Swap lets us efficiently implement many basic data structures such as stacks, queues, and linked lists because we can atomically update both a pointer and the location being pointed to; Section 2.3, p. 2 of Massalin]:

a linked-list of nodes encoding of a group of zero or more values [Section 3.3, General Linked Lists, p. 5 of Massalin]; and

linearizable operations defined to implement insert and remove operations on the group [Compare-and-Swap to perform linked-list insert and delete from the head of the list, Section 3.3, p. 5, second paragraph; in the middle of the list, we can achieve the same effect by deleting a node only when the permanence of the previous node is guaranteed; Section 3.3, p. 6 of Massalin], wherein concurrent execution of the linearizable operations [col. 8, lines 3 – 13 of Srinivas] is mediated using a first synchronization primitive [set the mark at the mark at the same time we enter the node using a two-word Compare-and-Swap; Section 3.3, p. 6 of Massalin] to encode a marked node indication signifying logical deletion of a corresponding one of the values from the group [(1) mark the nodes to be deleted and leave them in the list (2) if the previous node is not marked for deletion, sit on it and delete the original node marked for deletion; p. 6, Section 3.3 of Massalin] where the first synchronization primitive atomically examines [CAS, as an atomic operation; col. 8, lines 3 – 13 of Srinivas] and updates a single target, the updating being conditional on the examination [col. 7, line 60 – col. 8, line 3 of Srinivas].

15. As to claim 25, this is a product claim that corresponds to claim 1; note the rejection to claim above, which also meets this product claim.

16. As to claim 29, this is an apparatus claim that corresponds to method claim 16; see the rejection to claim 16 above, which also meets this apparatus claim. As to the additional limitations, Massalin teaches plural processors [multiprocessor OS kernel

using only lock-free synchronization methods based on Compare-and-Swap; see abstract], one or more data stores [quajects, chunks of code with data structures; Section 3, p. 3], and means for traversing the encoded group without use of an atomic operation [separate the run-queue traversal...from the queue element update; p. 3, 4th full paragraph].

17. As to claim 2, Massalin teaches physically excise the node corresponding to the logically deleted value [if the previous node is not marked for deletion, sit on it and delete the original node marked for deletion; p. 6, Section 3.3].

18. As to claim 3, Massalin as modified teaches compare and swap (CAS) operations and load/store-and-compare operations [col. 4, lines 13 – 29 of Srinivas].

19. As to claim 4, Massalin as modified teaches reclamation of storage associated with the excised node is independent of the linearizable [col. 6, lines 1 – 18 of Srinivas] operations [separate the run-queue traversal (done lock-free, safely and concurrently) from the queue element update (done locally); p. 3, 4th full paragraph of Massalin].

20. As to claim 5, Massalin teaches the linked-list of nodes is free of reference count storage for coordination of garbage collection [simplify the implementation to use a binary marker instead of counters; p. 6, 1st full paragraph].

21. As to claim 6, Massalin teaches traversal of the concurrent shared object is without atomic update of a garbage collection coordination store [separate the run-queue traversal (done lock-free, safely and concurrently) from the queue element update (done locally); p. 3, 4th full paragraph].

22. As to claim 7, Massalin teaches successful completion of an insertion into the group requires at most one atomic update of the concurrent shared object [insert reads the address of the list's first element into a private variable...copies it into the link field of

the new element to be inserted, and then uses Compare-and-Swap to atomically update the list's head pointer if it had not been changed since the initial read; Section 3.3, p. 5], successful completion of a deletion from the group requires, at most, two atomic updates of the concurrent shared object [do this in two steps: (1) mark the nodes to be deleted and leave them in the list (2) if the previous node is not marked for deletion, sit on it and delete the original node marked for deletion; p. 6, Section 3.3], and traversal of the concurrent shared object is without atomic update of the concurrent shared object [separate the run-queue traversal (done lock-free, safely and concurrently) from the queue element update (done locally); p. 3, 4th full paragraph].

23. As to claim 8, the node corresponding to the logically deleted value is physically excised from the linked-list by an execution sequence corresponds to one of:

an instance of the remove operation that performed the logical deletion [since step 2 may require going back the list an arbitrary number of nodes, usually we do the step 2 the next time we traverse the list to avoid the overhead of traversing the list just for deletion; p. 6, Section 3.3];

an instance of the remove operation that did not performed the logical deletion [if the previous node is not marked for deletion, sit on it and delete the original node marked for deletion; p. 6, Section 3.3]; and

an instance of the insert operation [insert reads the address of the list's first element into a private variable...uses Compare-and-Swap to atomically update the list's head pointer if it had not been changed since the initial read; Section 3.3, p. 5].

24. As to claim 10, Massalin teaches the values of the group are stored in respective ones of the nodes [if the shared data is larger than two words, then we try to encapsulate it in on of the lock-free objects; p. 3, 2nd full paragraph].

25. As to claims 11 and 12, Massalin teaches the values of the group are represented in storage identified by respective ones of the nodes [read the current value

of the stack pointer into a private variable, de-reference it to get the top item on the stack, and increment the stack pointer into a private variable; Section 3.1, p. 4].

26. As to claim 13, Massalin teaches the marked node indication includes a distinguishing pointer value [simplify the implementation to use a binary marker instead of counters; p. 6, 1st full paragraph].

27. As to claim 14, Massalin teaches the marked node indication includes a distinguishing bit value [simplify the implementation to use a binary marker instead of counters; p. 6, 1st full paragraph] in an otherwise unused portion of a next node pointer of the logically deleted node [set the mark at the time we enter the node using a two-word Compare-and-Swap; p. 6, 1st full paragraph].

28. As to claim 15, Massalin teaches respective next node pointers of those of the nodes corresponding to current values of the group directly reference respective other ones of the nodes [delete operation is "safe" if the deleted node's link pointers continue to be valid, i.e., pointing to nodes that eventually take it back to the main list where the Compare-and-Swap will detect the change; Section 3.3, p. 6], and the marked node indication includes a distinguishing additional level of indirection between the next node pointer of the logically deleted node and a respective other one of the nodes [two-word Compare-and-Swap can guarantee safety by simultaneously checking the previous node's pointer; Section 3.3, p. 6].

29. As to claim 17, this is rejected for the same reasons as claim 8 above.

30. As to claim 18, Massalin teaches the logical deletion is performed as part of a deletion operation operating upon the value [(1) mark the nodes to be deleted and leave them in the list; p. 6, Section 3.3], and the marked node excision is performed as part of another operation [(2) if the previous node is not marked for deletion, sit on it and delete the original node marked for deletion; p. 6, Section 3.3].

31. As to claims 19 and 20, these are rejected for the same reasons as claim 8 above.

32. As to claim 21, this is rejected for the same reasons as claim 3 above.

33. As to claim 22, Massalin teaches after the logical deletion but before the marked node excision, traversing, as part of an access operation, the linked-list including the marked node [since step 2 may require going back the list an arbitrary number of nodes, usually we do the step 2 the next time we traverse the list to avoid the overhead of traversing the list just for deletion; p. 6, Section 3.3].

34. As to claims 26 and 27, see the rejection to claim 1 above.

35. As to claim 28, Massalin teaches at least one computer readable medium is selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium [2.5 MB no-wait state main memory, 390 MB hard disk; p. 17, Section A.2].

36. As to claim 30, this is rejected for the same reason as claim 3 above.

37. **Claims 9, 23 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Massalin and Srinivas in view of U.S. Patent NO. 6,581,063 to Kirkman [cited in the previous office action].**

38. As to claim 9, Massalin as modified teaches managing access to a linked-list of nodes [Section 3.3, p. 5 – 6 of Massalin], but does not teach a find operation.

However, Kirkman teaches linked-list maintaining including an insert operation [col. 6, lines 40 – 63], a delete operation [remove function; col. 6, line 64 – col. 7, line

20] and a find operation [searching is necessary, there are not an excessive number of inactive objects to search through; col. 72, lines 15 – 35].

39. It would have been obvious to a person of ordinarily skilled in the art at the time of the invention to apply the teaching of a find operation as taught by Kirkman to the invention of Massalin as modified because this provides an pre-defined search capability that returns a requested element of the linked list without additional code for traversing the link-list to locate the requested element.

40. As to claims 23 and 24, Massalin as modified teaches the ordered set is organized in increasing value order [link may contain a textual data field such as a name, where the links are to be ordered such that the textual data fields are arranged in alphabetical order; col. 76, lines 13 – 27 of Kirkman], and the remove operation is selective for a value, if any, of the group greater than or equal to the specified value [find the link which is yet to be removed…The link which is to be removed is located; col. 27, lines 40 – 67 of Kirkman].

CONTACT INFORMATION

41. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (571) 272-3768. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on 571-272-3718. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Li B. Zhen
Examiner
Art Unit 2194

lbz

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER